Generating and Summarizing Explanations for Linked Data

Rakebul Hasan

Generating and Summarizing Explanations for Linked Data

Rakebul Hasan

INRIA Sophia Antipolis, Wimmics, 2004 route des Lucioles - B.P. 93,
06902 Sophia-Antipolis Cedex, France,
hasan.rakebul@inria.fr

Abstract. Linked Data consumers may need explanations for debugging or understanding the reasoning behind producing the data. They may need the possibility to transform long explanations into more understandable short explanations. In this paper, we discuss an approach to explain reasoning over Linked Data. We introduce a vocabulary to describe explanation related metadata and we discuss how publishing these metadata as Linked Data enables explaining reasoning over Linked Data. Finally, we present an approach to summarize these explanations taking into account user specified explanation filtering criteria.

Keywords: Explanation, summarization, Linked Data

1 Introduction

In the recent years, we have seen a growth of publishing Linked Data from community driven efforts, governmental bodies, social networking sites, scientific communities, and corporate bodies [7]. These data publishers from various domains publish their data in an interlinked fashion[1] using vocabularies defined in RDFS/OWL. This presents opportunities for large-scale data integration and reasoning over cross-domain data. In such a distributed scenario, consumers of these data may need explanations for debugging or understanding the reasoning behind producing the data; they may need the possibility to transform long explanations into more understandable short explanations [19]. Much of the previous work on explanations for the Semantic Web does not address explanation in a distributed environment [12]. The Inference Web [19] approach proposes a centralized registry based solution for publishing explanation metadata from distributed reasoners. We propose a decentralized solution to this problem. In essence, we discuss how to explain Linked Data in a decentralized fashion and how to summarize the explanations.

To explain Linked Data in a decentralized fashion, we publish explanation related metadata as Linked Data. In this approach, there is no constrain to publish the explanation metadata in a centralized location like in the Inference Web approach. To generate explanations, we retrieve the metatada by following

[1] See http://richard.cyganiak.de/2007/10/lod/ for a graph linking these datasets.

their dereferenceable URIs and present them in a human understandable form. For publishing explanation related metadata, we present a vocabulary to describe explanation metadata and guidelines to publish these metadata as Linked Data. To provide short explanations, we summarize the explanations by centrality, coherence, abstractness, and concept filtering.

The structure of the rest of this paper is as follows: in section 2, we present the related work. In section 3, we discuss how to represent and generate explanations for Linked Data. In section 4, we present our approach to summarize explanations. In section 5, we evaluate our summarization approach. Finally, we conclude and discuss the future work in section 6.

2 Related Work

Inference Web [19–21] is an explanation infrastructure which addresses explanation requirements of web services discovery, policy engines, first order logic theorem provers, task execution, and text analytics. Information manipulation traces of these various kinds of systems are encoded using Proof Markup Language (PML) [25]. Inference Web provides a set of software tools and services for building, presenting, maintaining, and manipulating PML proofs. PML is an explanation interlingua consisting of three OWL ontologies: PML provenance ontology (PML-P), PML justification ontology (PML-J), and PML trust ontology (PML-T). Inference Web authors define justification as a logical reasoning step, or any kind of computation process, or a factual assertion or assumption. Inference Web proposes a centralized registry based solution for publishing explanation metadata from distributed reasoners. In contrast, we propose a decentralized solution to address explanations in the distributed setting of Linked Data. The WIQA (Web Information Quality Assessment) framework [6] provides explanations of information filtering process for supporting information consumers in their trust decisions. WIQA exposes its explanations in RDF using the Explanation (EXPL) Vocabulary[2]. Forcher *et al.* [11] present the explanation-aware semantic search engine called KOIOS. The keyword search result explanations include information on how keywords are mapped to concepts and how concepts are connected. KOIOS uses a set of ontologies to formally describe the content of explanations in RDF. Both WIQA and KOIOS provide application specific explanations which include process descriptions of specific algorithms. In contrast, our explanations are suitable for generic Linked Data scenarios. Horridge *et al.* [14] present justification based explanation techniques. The authors define a justification for an entailment in an ontology as "a minimal subset of the ontology that is sufficient for the entailment to hold". The authors present laconic and precise justifications which are fine-grained justifications consisting of axioms with no superfluous part. The authors present an optimized algorithm to compute laconic justifications showing the feasibility of computing laconic justifications and precise justifications in practice. We do not focus on the theoretical

[2] http://www4.wiwiss.fu-berlin.de/bizer/triqlp/

aspects of the justifications such as the minimal parts of axioms in a justification which are required to hold an entailment. Rather, we focus on the aspects related to publishing and consuming explanation metadata in a distributed environment. Other notable works on explanations in the Semantic Web literature include OntoNova [3] and Knowledge in a Wiki (KiWi) [17]. OntoNova and KiWi provide explanations of their reasoning. However, they do not represent their explanation metadata using standard data formats. This is an undesirable situation for Linked Data scenarios because data consumers would not be able to process such non standard explanation metadata.

To the best of our knowledge, there is no comparable published work on summarizing explanations in the Semantic Web literature. But researchers have studied ontology summarization. RDF Sentence graph based summarization [27] extracts RDF sentences based on centrality measures. Our work has a similar approach to sentence graph summarization approach. However, we define new measures for summarizing explanations. Peroni *et al.* [23] discuss how to identify key concepts in an ontology. They draw summarization criteria from cognitive science (natural categories), network topology (density and coverage), and lexical statistics (term popularity). Alani *et al.* [2] discuss shrinking an ontology by analyzing the usage of the ontology. Alani *et al.* analyze the query log against an ontology to understand the important parts of the ontology. Peroni *et al.* and Alani *et al.* focus on a concept level summarization of ontologies. In contrast, our focus is on statement level summarization.

3 Explaining Linked Data

We follow the Linked Data principles [5] to publish our explanation metadata. We describe these metadata using our proposed vocabulary *Ratio4TA*[3]. We generate explanations by retrieving the explanation metatada by following their dereferenceable URIs and presenting them in a human understandable form.

3.1 Representing and Publishing Explanation Metadata

Ratio4TA (inter*r*linked explan*ati*ons *for t*riple *a*ssertions) is an OWL ontology for describing explanation metadata. *Ratio4TA* extends the W3C PROV Ontology[4]. This promotes interoperability by enabling data consumers process explanation metadata according to W3C PROV standards. Consumers of these explanation metadata can use their preferred tools to present and visualize explanations. Figure 1 shows the core concepts and relations of *Ratio4TA*. They allow describing data, reasoning processes, results, data derivations, rules, and software applications. The *ExplanationBundle* concept allows to define named graph containers for RDF statements representing explanation metadata.

We publish the explanation metadata as Linked Data. This means that all the resources in our explanation metadata have dereferenceable HTTP URIs.

[3] `http://ns.inria.fr/ratio4ta/`
[4] `http://www.w3.org/TR/prov-o/`

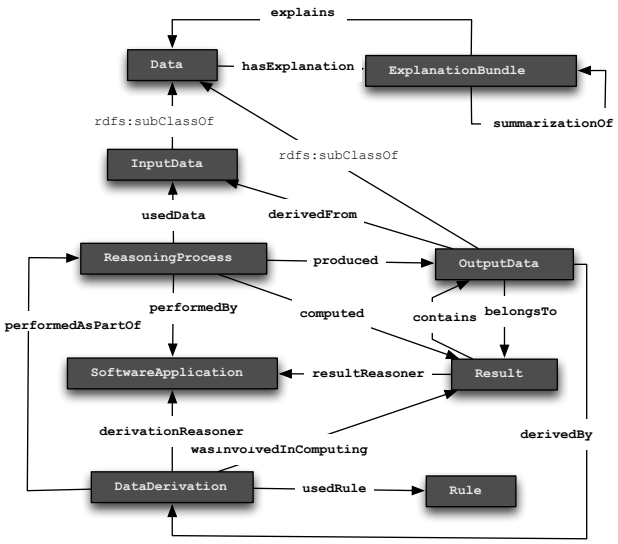

Fig. 1: The core classes and properties of *Ratio4TA*.

We avoid using blank nodes to keep the resources globally dereferenceable. We use the named graph mechanism [8] to make statements about RDF triples. Using named graph allows us to associate explanation metadata for data with different level of granularity – explanation metadata for a triple or a graph containing more than one triple. Furthermore, we use named graphs to group together explanation metadata and make the metadata for an explanation referenceable by a single URI. Listing 1.1 shows an extract of an explanation described using *Ratio4TA* in TriG [8] notation. The example in this listing shows the explanation metadata for the derived triple *lodapp:data1*. The named graph *lodapp:explanation1* contains the explanation metadata. The metadata include links to the reasoning process, the input data, the rule, the software application, and the result to which the derivation contributes.

Listing 1.1: Extract from the explanation metadata for a derivation

```
# Explanation Metadata
lodapp:explanation1 {
  lodapp:explanation1 r4ta:explains lodapp:data1.
  # Type declarations
  lodapp:explanation1 rdf:type r4ta:ExplanationBundle.
  lodapp:corese rdf:type r4ta:SoftwareApplication.
  ....
  ....
  # Reasoning process
  lodapp:reasoningProcess1 r4ta:performedBy lodapp:corese;
      r4ta:usedData lodapp:inputData1;
      r4ta:usedData lodapp:inputData2;
      r4ta:computed lodapp:result1;
      r4ta:produced lodapp:data1.
  # Computed result
  lodapp:result1 r4ta:resultReasoner lodapp:corese .
  # Output data
  lodapp:data1 r4ta:derivedFrom lodapp:inputData1;
      r4ta:derivedFrom lodapp:inputData2;
      r4ta:belongsTo lodapp:result1;
      r4ta:derivedBy lodapp:derivation1.
  # Data derivation
  lodapp:derivation1 r4ta:usedRule lodapp:geoFeatureRule;
      r4ta:wasInvolvedInComputing lodapp:result1;
      r4ta:derivationReasoner lodapp:corese;
      r4ta:performedAsPartOf lodapp:reasoningProcess1.
}
# Derived data
lodapp:data1 {
  dbpedia:Philadelphia gn:parentFeature geonames:5205788.
}
# Dbpedia data
lodapp:inputData1 {
  dbpedia:Philadelphia owl:sameAs geonames:4560349 .
}
# GeoNames data
lodapp:inputData2 {
  geonames:4560349 gn:parentFeature geonames:5205788.
}
```

Why a New Ontology? Proof Markup Language (PML) [25] and the AIR Justification Ontology (AIRJ) [16] are important previous works on representing explanation metadata. PML allows describing provenance metadata, justifications for derivations of conclusions, and trust related metadata. Additionally, a light weight variant of PML known as PML-Lite [24] presents a simple subset of PML. AIRJ extends PML-Lite and provides primitives to represent the different events and the operations performed by reasoners. PML and AIRJ use RDF container concepts. RDF containers use blank nodes to connect a sequence of items [1]. However, as a common practice, blank nodes are avoided while publishing Linked Data [13]. It is not possible to make statements about blank nodes as they do not have identifiers. Therefore, blank nodes make data integration harder in the global dataspace of Linked Data. Additionally, the existing ontologies do not use any common data interchanging standard such as W3C PROV-O. This makes it hard for applications across the web to make sense of the explanation metadata.

3.2 Generating Explanation

We generate explanations from the published explanation metadata by recursively following the links between the involved explanation metadata and the data they describe. For a derived RDF statement *dst*, we crawl through the related metadata with a maximum depth limit and collect the set of explanation meta statements, and the set of RDF statements from which the derived RDF statement *dst* is derived. In the remaining for this paper, we refer to the derived RDF statement (the initial *dst*) that we are explaining as the root statement rs. We refer to the set of RDF statements from which rs is derived as knowledge statements KST. The RDF knowledge graph KG is the graph formed by union of KST and the root statement: $KG = RDFGraph(KST \cup rs)$. We generate natural language descriptions from the RDF statements in KG (using *rdfs:label* property values) and present them as explanations for human end-users. Figure 2 shows an example of our explanation for a derived statement that "Bob is a British Scientist". Each derivation contains a link to the natural language representation of the used rule. Although this kind of explanations with the details

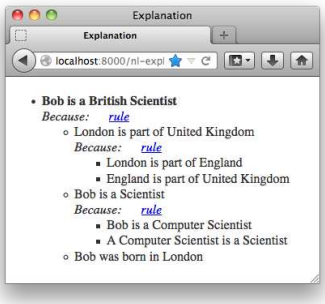

Fig. 2: Full explanation.

Fig. 3: Summarized explanation.

of all the steps may be useful for expert users, they may overwhelm non-expert users with too much information [3, 12, 21]. We provide summarized explanations to address this problem. Figure 3 shows an example of a summarized explanation for "Bob is a British Scientist". Users can switch to the full explanation by clicking on the "more details" link. In the next section, we discuss our approach to summarizing explanations.

4 Summarizing Explanations

In [3, 12, 21], researchers discuss the importance of providing short explanations rather than overwhelming the end-users with too much information. The authors of [3] also discuss filtering information in explanations in order to provide more relevant explanations. We propose an approach to summarizing explanations taking into account user specified filtering criteria. More formally, let $KG = (R, T)$ be an RDF knowledge graph, where R is the set of resources and literals and T is the set of RDF statements. Let rs be the root statement (therefore the knowledge statements $KST = T \setminus rs$). We provide summarized explanations by summarizing RDF statements from KST. We use the term "oriented graph" to refer to KG throughout the paper. Our summarization approach includes first a ranking step and then a re-ranking step. It is important to note that our summarized explanations may not always conform to the correctness of deductions from a logical point of view. Our summarized explanations are not aimed at explaining the correct deduction steps. Rather the aim is to provide a short overview of the background information used in a deduction. We describe below the measures we use for summarizing explanations.

4.1 Measures for Ranking

We rank the statements in KST based on their scores we compute using our summarization measures. The scores are normalized and range from 0.0 to 1.0. A higher score for a statement means that the statement is more suitable for a summary. Taking n statements, where $n <\mid KST \mid$, or statements with scores greater than a threshold value will give a summarized list of statements which can explain rs. For the ranking step, we compute the scores by using three measures: salience (S_{SL}), similarity (S_{SM}), and abstractness (S_{AB})

Salient RDF Statements The salience of an RDF statement indicates the importance of the RDF statement in the oriented graph. We use normalized degree centrality, $C_{DN}(v)$, to compute salience of RDF statements. Degree centrality of a vertex in a graph is the number of links the vertex has. We compute the salience $S_{SL}(i)$ of an RDF statement i using (1).

$$S_{SL}(i) = \theta_1 \times C_{DN}(subjectOf(i)) + \theta_2 \times C_{DN}(objectOf(i)) \quad (1)$$

In (1), $\sum_i \theta_i = 1$ and $\forall_i : \theta_i \geq 0$ *i.e.* we take the weighted average of the normalized degree centrality of the subject and the object of the RDF statement

i. The $subjectOf(i)$ and the $objectOf(i)$ functions return respectively the subject resource and the object resource of the RDF statement i. We did not use the centrality of the predicate of statement while computing S_{SL} because we wanted an importance score representing the importance of the information in a statement but not the importance of the relation between the information. The centrality values of predicates in a RDF graph often do not change as they are directly used from the schemata. In contrast, every new RDF statement changes the centrality values of its subject and object.

Similar RDF Statements The consumers of our explanations can specify a set of classes, FL, as their filtering criteria, where $FL \subseteq SC$ and SC is the set of all classes in the schemata used to describe KG. We rank the more similar statements to the concepts given in filtering criteria higher. We use the approximate query solving feature [9] of Corese[5] to compute similarity. The approximate query solving feature is a semantic distance-based similarity feature to compute conceptual similarity between two classes in a schema. For a statement i and a set of classes as filtering criteria FL, we compute similarity $S_{SM}(i, FL)$ using (2).

$$\begin{aligned} S_{SM}(i, FL) = \theta_1 \times similarity_{node}(subjectOf(i), FL) \\ + \theta_2 \times similarity_{node}(predicateOf(i), FL) \\ + \theta_3 \times similarity_{node}(objectOf(i), FL) \end{aligned} \tag{2}$$

The function $predicateOf(i)$ returns the predicate of the statement i. We compute $similarity_{node}(j, FL)$ where $j \in R \cup SC$ as following:

$$similarity_{node}(j, FL) = \begin{cases} similarity_{type}(\{j\}, FL) & \text{if } j \in SC \\ similarity_{type}(typesOf(j), FL) & \text{if } j \notin SC \end{cases} \tag{3}$$

In (3), for the case $j \in SC$, we compute the similarity between the class j and the set of classes in FL. For the case $j \notin SC$, we compute the similarity between the set of classes of which j is an instance and the set of classes in FL. The $similarity_{type}$ function takes as arguments a set of classes $TP \in SC$ and the set of filtering criteria FL, and returns the similarity value between them. The $typesOf(j)$ function for a resource $j \in R$ returns the set of classes of which j is an instance. The $similarity_{type}$ function in (4a) computes its value by taking the average of all the values of $maxSimilarity_{type}(m, TP)$ where $m \in FL$ and $TP \in SC$. The $maxSimilarity_{type}$ function in (4b) returns the maximum similarity value between a class m and all the classes in TP. This is to ensure that when a resource is an instance of multiple classes, we filter it by the class which is more similar to the filtering criteria. The $similarity_{type}$ function calculates a combined similarity score of TP with respect to all the classes in FL. Again, we

[5] http://wimmics.inria.fr/corese

consider the weighted average, and therefore $\sum_i \theta_i = 1$ and $\forall_i : \theta_i \geq 0$ in (2).

$$similarity_{type}(TP, FL) = \frac{\sum_{m \in FL} maxSimilarity_{type}(m, TP)}{\mid FL \mid} \tag{4a}$$

$$maxSimilarity_{type}(m, TP) = \max_{n \in TP} (similarity_{corese}(m, n)) : \tag{4b}$$

For a class $m \in FL$ and a class $n \in TP$, $similarity_{corese}(m, n)$ computes the similarity score between class m and n ranging from 0.0 to 1.0 using SPARQL similarity extension of Corese. A value of 1.0 represent exact match and a value of 0.0 represents completely not similar. The S_{SM} score for a statement indicates the similarity of the information in the statement to the information specified in FL.

Abstract Statements We consider a statement that is close to the root, rs, in corresponding proof tree is more abstract than a statement that is far from the root rs. We define the distance of a node in the proof tree from the root node as the level of the tree to which the node belongs. The root node belongs to level one in the proof tree. The root node is derived from the nodes in level two. A node in level two is derived from the nodes in level three, and so on. For a statement $i \in KST$, we compute the abstraction score $S_{AB}(i)$ using (5).

$$S_{AB}(i) = \frac{1}{level(i)} \tag{5}$$

The function $level(i)$ returns the proof tree level to which the statement i belongs. The $S_{AB}(i)$ measure gives a value greater than 0.0 and less than or equal to 1.0, where a smaller value means less abstract and a larger value means more abstract.

4.2 Measures for Re-Ranking

We use two more measures to improve the rankings produced by the combinations of three measures we presented so far.

Subtree Weight in Proof Tree For a subtree of the proof tree with root i, we compute the subtree weight of the statement i by taking the average score of all the statements in that subtree.

$$score_{ST}(i) = \frac{\sum_{j \in subtree(i)} score(j)}{\mid subtree(i) \mid} \tag{6}$$

The $subtree(i)$ function returns the RDF statements from the subtree of proof tree with root i. The $score(j)$ for a statement j here can be computed by combinations of the measures we present in section 4.1. We discuss more about how to combine the different measures in section 5.

Coherence Previous works in text summarization [10] and ontology summarization [27] have shown that coherent information are desirable in summaries. We consider an RDF statement x to be coherent to an RDF statement y if x is directly derived from y. Let RL be a ranked list of RDF statements; S be a list of already selected RDF statements in the summary; i be the next RDF statement to be selected in S. We re-rank RL by repeatedly selecting next i with $\mid RL \mid$ repetitions using (7).

$$i = \underset{j \in RL \setminus S}{\arg\max} \left(\lambda_1 \times score(j) + \lambda_2 \times reward(j, S)\right) \tag{7}$$

Again, the $score(j)$ for a statement j here can be computed by combinations of the measures we presented before. We take the weighted average of $score(j)$ and $reward(j, S)$ in (7), therefore $\sum_i \lambda_i = 1$ and $\forall_i : \lambda_i \geq 0$.

$$reward(j, S) = 1 - \frac{coherent(S)}{coherent(S \cup j)} \tag{8}$$

As (8) shows, the $reward$ score of a statement j is the amount of potential contribution value – ranging from 0.0 to 1.0 – to the total coherence of the summary if j is added to S. The function $coherent(S)$ in (8) returns the number of coherent statements in the summary S.

5 Evaluation

Ontology summarization [18] and text summarization [10, 26] technologies are evaluated by measuring agreements between human-generated summaries – known as "ground truths" – and automatically generated summaries. We obtained our ground truths by surveying 24 people: 17 computer scientists, 1 chemist, 1 social scientist, 1 mathematician, 1 journalist, 1 psychologist, 1 biologist, and 1 business administrator. 18 participants in our survey had knowledge of RDF and 6 participants did not have any knowledge of RDF. The ages of the participants range from 22 to 59. 20 participants were male and 4 were female. The explanations, the questionnaires, the responses, and the results of the evaluation are publicly available online[6]. We selected a subset of geographical locations from GeoNames[7] and a subset of artists, events, and places from DBPedia[8], then derived new information from these selected subsets. Our ad-hoc reasoner infers new RDF statements with respect to RDFS type propagation; and owl:sameAs and transitivity of the parentFeature property of GeoNames schema. In addition, the reasoner generates explanations for each derivation it performs. We used three test cases – three queries with their results along with the explanations for the results. Each query result is an inferred statement by our reasoner. Each test case has two scenarios: without filtering criteria FL, and with filtering

[6] http://ns.inria.fr/ratio4ta/sm/
[7] http://www.geonames.org/
[8] http://dbpedia.org/

criteria FL. Each participant answered questions for one test case. We randomly assigned a test case to a participant. We ask the participants to rate, from a scale of 1 to 5, the need for each of the statements in the explanation. For, the scenario with filtering criteria FL, we give the query, the answer, and the explanation but with a user's filtering criteria class taken from the schemata used in the reasoning process. The ratings of the explanation statements are our ground truths. We compute the ground truth rankings of explanation statements by ordering them by their rating values.

5.1 Comparing Summarization Measures

We evaluate different combinations of the summarization measures we define. In equation (9), we compute $score_{S_{SL}}(i)$ for a statement i considering salience of the statement. We always include S_{SL} in our measure combinations. The motivation is to first include the salient statements in a summary and then find the statements with other measure combination scores (*e.g.* S_{AB} or S_{SM} or S_{AB}+S_{SM}) in those salient statements. Equations (10), (11), and (12) show three more combinations of measures that we consider for our evaluation. In (10), we compute $score_{SL+AB}(i)$ for a statement i considering salience and abstractness of the statement. In (11), we compute the $score_{SL+SM}(i)$ for a statement i considering the salience (S_{SL}), and the similarity (S_{SM}) with respect to user's filtering criteria FL. In (12), we compute $score_{SL+AB+SM}(i)$ for a statement i considering the salience (S_{SL}), the abstractness (S_{AB}), and the similarity (S_{SM}) with respect to user's filtering criteria FL.

$$score_{SL}(i) = S_{SL}(i) \tag{9}$$

$$score_{SL+AB}(i) = \lambda_1 \times S_{SL}(i) + \lambda_2 \times S_{AB}(i) \tag{10}$$

$$score_{SL+SM}(i) = \lambda_1 \times S_{SL}(i) + \lambda_2 \times S_{SM}(i, FL) \tag{11}$$

$$\begin{aligned} score_{SL+AB+SM}(i) = \lambda_1 \times S_{SL}(i) + \lambda_2 \times S_{AB}(i) \\ + \lambda_3 \times S_{SM}(i, FL) \end{aligned} \tag{12}$$

These combinations are combinations of ranking measures we present in section 4.1. For re-ranking, we first compute the score using any of (9), (10), (11), and (12), then we re-rank using (6), or (7). In remaining of this paper, we denote subtree weight measure as S_{ST}, and coherence measure as S_{CO}. For the scenario without FL, we compare our summaries to sentence graph summarization [27] – denoted as S_{SG}. As the authors of sentence graph summarization approach suggest, we use 0.8 as the navigational preference p parameter value. We implemented sentence graph summarization using degree centrality as the authors found degree centrality performs better than other centrality measures in general, and for its simplicity. We do not consider sentence graph summarization for the scenarios with FL because sentence graph summarization does not have a feature for filtering information using ontology concepts as filtering criteria.

In (10), (11), and (12), $\sum_i \lambda_i = 1$ and $\forall_i : \lambda_i \geq 0$. Thus we take the weighted averages of the measure combinations. For this evaluation, we use equal

weights in (10), (11), (12), (1), (2), and (7). Therefore, we set $\forall_i : \lambda_i = \frac{1}{N_\lambda}$ in (10), (11), (12), and (7) where N_λ = number of λ parameters in the corresponding equations; and $\forall_i : \theta_i = \frac{1}{N_\theta}$ in (1), and (2) where N_θ = number of θ parameters in the corresponding equations. However, one can use parameter estimation techniques for finding the optimal parameter values.

5.2 Analysis of Ground Truths

We use cosine similarity to measure the agreements between rating vectors. Cosine similarity values in positive space are in the interval 0 to 1. Table 1 shows the total average agreement measured by cosine similarity and standard deviations for two scenarios – without filtering criteria FL and with filtering criteria FL. The average agreements for both the scenarios are more than 0.8

	avg.	std. dev.
Without FL	0.836	0.048
With FL	0.835	0.065

Table 1: Average agreements between ratings measured by cosine similarity.

which is considerably high. However, the standard deviation is higher for the scenario with FL. The reason for this higher standard deviation is that the participants had to consider the highly subjective [4] factor of similarity and therefore their ratings had more variance for the scenario with FL.

5.3 Evaluating the Rankings

We use normalized discounted cumulative gain to evaluate ranking quality. Discounted cumulative gain (DCG) [15, 22] measures the quality of results of an information retrieval system in a ranked list. DCG assumes that judges have graded each item in a list of results. Using these grades, DCG measures the usefulness, or gain, of a ranked list of results. DCG penalizes high quality results appearing lower in a ranked list of results. Normalized discounted cumulative Gain ($nDCG$) allows to calculate and compare this measure across multiple lists of results where each of the lists might have different length. $nDCG$ values are in the interval 0.0 to 1.0. An $nDCG_p$ value of 1.0 means that the ranking is perfect at position p with respect to the ideal ranking – ranking based on grades. The $nDCG_p$ value 0.0 means that the ranking is completely imperfect at position p with respect to the ideal ranking. In our study, the average of ratings by all the survey participants for a statement s is the grade for the statement s. Figure 4 shows the average $nDCG$ values of the three test cases for different rankings by different measure combinations. The *x-axis* represents ranks and the *y-axis* represents $nDCG$. We plot 21 ranks in the *x-axis* because the shortest explanation among the three test cases had 21 statements. For the scenario without FL (the figure on the left), the measure combinations $S_{SL}+S_{AB}+S_{CO}$, $S_{SL}+S_{AB}+S_{ST}$, and $S_{SL}+S_{AB}+S_{ST}+S_{CO}$ produce more closer rankings to the ground truth rankings. For the scenario with FL (the figure on the right), the same three measure combinations with added S_{SM} measure have the best $nDCG$ values.

This means that the participants consider central (with respect to the oriented graph and the proof tree), abstract, and coherent information as necessary information in explanation summaries for the scenario without FL. This also holds for the scenario with FL with the added observation that the participants also consider similar information as necessary information. The $nDCG$ values for these measure combinations are higher than 0.9 for all ranks. This means that the rankings by these measure combinations are highly similar to the ground truth rankings. In contrast, the sentence graph summarization ranking has low $nDCG$ values compared to all the other rankings for the scenario without FL. This shows that our explanation summarization algorithms produce much higher quality rankings than sentence graph summarization algorithm.

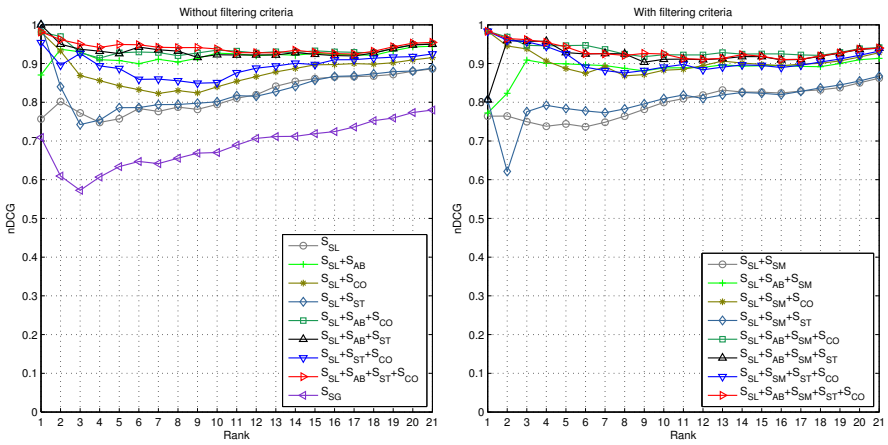

Fig. 4: Comparison of rankings.

5.4 Evaluating the Summaries

We evaluate the summaries using *Recall* and *Precision* composite scores as in text summarization [10]. *Recall* and *Precision* quantify how closely the algorithm generated summaries correspond to the human produced summaries. *Recall* reflects how many good statements the algorithm missed, and *Precision* reflects how many of the algorithm's selected statements are good. *F-score* is the composite measure of *Recall* and *Precision*. We use the basic *F-score* as in [26]: $\textit{F-score} = \frac{2 \times Precision \times Recall}{Precision + Recall}$. We measure *F-score* for summarized explanations with different compression ratios, CR, to evaluate summaries of different sizes. Compression ratio CR is the ratio of the size of the summarized explanation to the size of original explanation. We evaluate the summarized explanations produced by different measure combinations by comparing them to human generated summarized explanations (*i.e.* ground truth summarized explanations) using *F-score*. To generate the ground truth summarized explanation for an explanation, we include a statement in the ground truth summarized explanation if its rating is greater than or equal to the average rating of all the statements in

the original explanation. *F-scores* reflects the accuracy of automatically generated summaries with respect to the ground truth summary. A desirable situation would be a summarized explanation with high *F-score* and low CR. Figure 5 shows the average *F-scores* for different measure combinations for summaries with different sizes for the three test cases. The *x-axis* represents compression ratio CR. The *y-axis* represents *F-scores*. For the scenario without FL (the figure on the left), the best *F-score* is 0.72 when CR value is 0.33 by the measure combinations $S_{SL} + S_{AB} + S_{ST}$ and $S_{SL} + S_{AB} + S_{ST} + S_{CO}$. This is a desirable situation with a high *F-score* and low CR. The sentence graph summarization performs poorly with a best *F-score* value of 0.34 in the CR interval 0.05 to 0.3. This shows that our summarized explanations are more accurate than the summarized explanations generated by sentence graph summarization algorithm. For the scenario with FL (the figure on the right), the best *F-score* is 0.66 at CR values 0.53 and 0.55 by the measure combination$S_{SL} + S_{SM}$. However, the *F-score* 0.6 at CR value 0.3 by the measure combination $S_{SL} + S_{AB} + S_{SM} + S_{CO}$ is more desirable because the size of the summary is smaller. As expected, our summarization approach perform worse in the scenario with FL where we use S_{SM}. This is due to the fact that the survey participants had to consider the highly subjective factor of similarity.

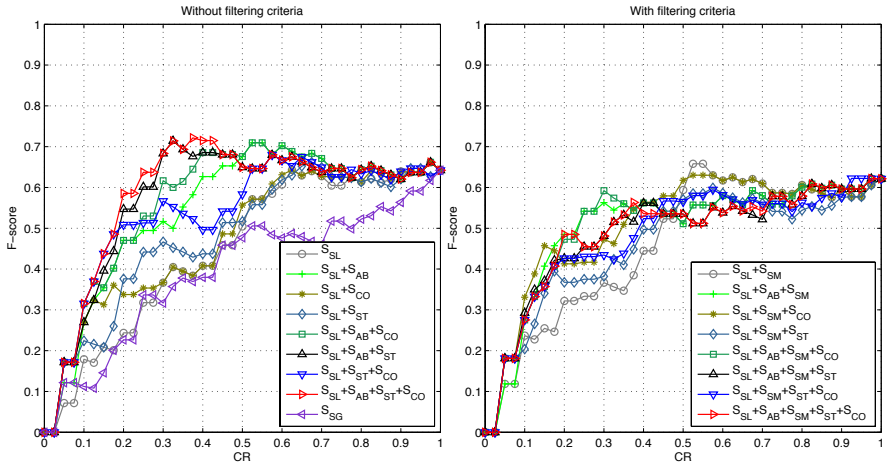

Fig. 5: Compression ratio (CR) vs *F-score*.

6 Conclusion and Future Work

In this paper, we discuss how to generate and summarize explanations for Linked Data. We present an ontology to describe explanation metadata and discuss publishing explanation metadata as Linked Data. In addition, we presented five summarization measures to summarize explanations. We evaluate different combinations of these measures. The evaluation shows that our approach produces high quality rankings for summarizing explanation statements. Our summarized

explanations are also highly accurate with *F-score* values ranging from 0.6 to 0.72 for small summaries. Our approach outperforms the sentence graph based ontology summarization approach.

In the future work, we would like to explore how we can effectively present explanations and summarized explanations using different kinds of user interfaces and user interactions. We would like to explore how we can effectively use the summarization rankings while presenting information in personalized scenarios. Finally, we are going to evaluate the impact of explanations and summarized explanations on end-users.

Acknowledgments: This work is supported by the ANR CONTINT program under the Kolflow project (ANR-2010-CORD-021-02).

References

1. RDF semantics. W3C recommendation (2004)
2. Alani, H., Harris, S., O'Neil, B.: Winnowing ontologies based on application use. In: Sure, Y., Domingue, J. (eds.) The Semantic Web: Research and Applications, LNCS, vol. 4011, pp. 185–199. Springer Berlin Heidelberg (2006)
3. Angele, J., Moench, E., Oppermann, H., Staab, S., Wenke, D.: Ontology-based query and answering in chemistry: Ontonova project halo. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) The Semantic Web - ISWC 2003, LNCS, vol. 2870, pp. 913–928. Springer Berlin / Heidelberg (2003)
4. Araújo, R., Pinto, H.S.: Towards semantics-based ontology similarity. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., He, B. (eds.) Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) at ISWC-2007/ASWC-2007 (2007)
5. Berners-Lee, T.: Linked Data. W3C Design Issues `http://www.w3.org/DesignIssues/LinkedData.html` (2006)
6. Bizer, C.: Quality-Driven Information Filtering in the Context of Web-Based Information Systems. Ph.D. thesis, Freie Universität Berlin (2007)
7. Bonatti, P., Hogan, A., Polleres, A., Sauro, L.: Robust and scalable linked data reasoning incorporating provenance and trust annotations. Web Semantics: Science, Services and Agents on the World Wide Web 9(2), 165–201 (2011)
8. Carroll, J.J., Bizer, C., Hayes, P., Stickler, P.: Named graphs, provenance and trust. In: Proceedings of the 14th international conference on World Wide Web. pp. 613–622. WWW '05, ACM, New York, NY, USA (2005)
9. Corby, O., Dieng-Kuntz, R., Gandon, F., Faron-Zucker, C.: Searching the Semantic Web: approximate query processing based on ontologies. Intelligent Systems, IEEE 21(1), 20–27 (2006)
10. Eduard, H.: Text summarization. In: Mitkov, R. (ed.) The Oxford Handbook of Computational Linguistics. Oxford University Press (2005)
11. Forcher, B., Sintek, M., Roth-Berghofer, T., Dengel, A.: Explanation-aware system design of the semantic search engine koios. In: Proc. of the the 5th Int'l. Workshop on Explanation-aware Computing (2010)
12. Hasan, R., Gandon, F.: A Brief Review of Explanation in the Semantic Web. Workshop on Explanation-aware Computing (ExaCt 2012), European Conference on Artificial Intelligence (ECAI 2012) (2012)

13. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool, 1st edn. (2011), http://linkeddatabook.com/
14. Horridge, M., Parsia, B., Sattler, U.: Laconic and precise justifications in OWL. In: Proc. of the 7th Int'l Conference on the Semantic Web. pp. 323–338. ISWC '08, Springer-Verlag (2008)
15. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst. 20(4), 422–446 (Oct 2002)
16. Kagal, L., Jacobi, I., Khandelwal, A.: Gasping for AIR – why we need linked rules and justifications on the semantic web. Tech. Rep. MIT-CSAIL-TR-2011-023, MIT (2011)
17. Kotowski, J., Bry, F.: A perfect match for reasoning, explanation and reason maintenance: OWL 2 RL and semantic wikis. In: Proc. of 5th Semantic Wiki Workshop (2010)
18. Li, N., Motta, E.: Evaluations of user-driven ontology summarization. In: Cimiano, P., Pinto, H. (eds.) Knowledge Engineering and Management by the Masses, LNCS, vol. 6317, pp. 544–553. Springer Berlin Heidelberg (2010)
19. McGuinness, D., Silva, P.: Infrastructure for web explanations. In: Fensel, D., Sycara, K., Mylopoulos, J. (eds.) The Semantic Web - ISWC 2003, LNCS, vol. 2870, pp. 113–129. Springer Berlin Heidelberg (2003)
20. McGuinness, D., Furtado, V., Pinheiro da Silva, P., Ding, L., Glass, A., Chang, C.: Explaining semantic web applications. In: Semantic Web Engineering in the Knowledge Society (2008)
21. McGuinness, D., Pinheiro da Silva, P.: Explaining answers from the semantic web: the inference web approach. Web Semantics: Science, Services and Agents on the World Wide Web 1(4), 397 – 413 (2004)
22. McSherry, F., Najork, M.: Computing information retrieval performance measures efficiently in the presence of tied scores. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R. (eds.) Advances in Information Retrieval, LNCS, vol. 4956, pp. 414–421. Springer Berlin Heidelberg (2008)
23. Peroni, S., Motta, E., d'Aquin, M.: Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures. In: Domingue, J., Anutariya, C. (eds.) The Semantic Web, LNCS, vol. 5367, pp. 242–256. Springer Berlin Heidelberg (2008)
24. Pinheiro da Silva, P., McGuinness, D., Del Rio, N., Ding, L.: Inference web in action: Lightweight use of the proof markup language. In: Proc. of the 7th Int'l Semantic Web Conference. pp. 847–860. ISWC '08 (2008)
25. Pinheiro da Silva, P., McGuinness, D., Fikes, R.: A proof markup language for semantic web services. Information Systems 31(4-5), 381–395 (2006)
26. Steinberger, J., Jezek, K.: Evaluation measures for text summarization. Computing and Informatics 28(2), 251–275 (2009)
27. Zhang, X., Cheng, G., Qu, Y.: Ontology summarization based on RDF sentence graph. In: Proceedings of the 16th international conference on World Wide Web. pp. 707–716. WWW '07, ACM, New York, NY, USA (2007)